This listing of claims will replace all prior versions, and listings, of claims in the application.

**Listing of Claims:**

1. (original) A distributed information processing system, comprising:

a client device interface adapted to receive requests for information from a plurality of remote devices;

a module manager adapted to receive and route said requests from said client device interface; and

a plurality of information modules,

wherein said information modules register with said module manager and module manager routes said request to an appropriate one of said plurality of information modules in accordance with a type of information requested.

2. (original) The distributed information processing system as recited in claim 1, wherein the requests to the client device interface are formatted as an HTML or plain-text formatted e-mail.

3. (original) The distributed information processing system as recited in claim 1, wherein the appropriate one of said plurality of information modules generates a response that is returned to said module manager, and wherein said module manager routes said response to said client interface device for delivery to a requestor.

4. (original) The distributed information processing system as recited in claim 1, wherein said requests and responses are formatted as serializable Java objects.

5. (original) The distributed information processing system as recited in claim 1, wherein said requests are made to said module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

6. (original) The distributed information processing system as recited in claim 1, wherein instances of said module manager are created each time a new request is received and discarded after the request has been handled.

7. (currently amended) The distributed information processing system as recited in claim 6, wherein instances of said module ~~managers~~ manager are stateless and multi-threaded.

8. (original) The distributed information processing system as recited in claim 1, wherein information modules are loaded locally and remotely, wherein local modules reside on a same physical device as said module manager, and wherein remote modules are located on other devices.

9. (original) The distributed information processing system as recited in claim 8, wherein communication between locally loaded modules and said module manager is accomplished via memory calls, object inheritance or inter-process communication.

10. (original) The distributed information processing system as recited in claim 8, wherein communication between remotely loaded modules and said module manager is accomplished via TCP/IP sockets.

11. (currently amended) The distributed information processing system as recited in claim 1, further comprising a subscription service that maintains a subscriber database, wherein information is sent by said information modules, and said subscription center is consulted to determine to which clients the information should be forwarded.

12. (currently amended) A method of receiving and responding to requests for ~~in an~~ information in a ~~distributing~~ distributed information processing system ~~comprising a client device interface adapted, a module manager, and a plurality of information modules~~, the method comprising:

receiving a request at ~~said~~ a client device interface;

forwarding said request to ~~said~~ a module manager;

consulting a registry of available information modules;

forwarding said request to an appropriate information module as determined in accordance with a type of information requested.

13. (currently amended) The method of claim 12, further comprising:

[processing the request at said appropriate information module;

generating a response that is returned to said module manager; and

routing said response o said client interface device for delivery to a requestor]

maintaining a list of supported services provided by each of said information modules; and

handling service collisions if plural information modules are capable of responding to said type of information such that only one information module processes said request.

14. (original) The method of claim 12, wherein said requests and responses are formatted as serializable Java objects.

15. (original) The method of claim 12, wherein said requests are made to said module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

16. (currently amended) The method of claim 12, ~~wherein said information processing system further comprises a subscription service that maintains a subscriber database,~~ said method further comprising:

[generating information in the form of a response at one of said information modules;

consulting ~~said~~ a subscriber database; and

forwarding said response to clients in accordance with information in said subscriber database]

creating an instance of said module manager upon receiving said request; and

discarding said instance after said response has been handled.

17. (new) A computer readable medium containing computer executable instructions for receiving and responding to requests for information in a distributed information processing system, said computer executable instructions for performing the steps of:

receiving a request at a client device interface;

forwarding said request to a module manager;

consulting a registry of available information modules;

forwarding said request to an appropriate information module as determined in accordance with a type of information requested.

18. (new) The computer readable medium of claim 17, further comprising computer executable instructions for performing the steps of:

maintaining a list of supported services provided by each of said information modules; and

handling service collisions if plural information modules are capable of responding to said type of information such that only one information module processes said request.

19. (new) The computer readable medium of claim 17, wherein said requests and responses are formatted as serializable Java objects.

20. (new) The computer readable medium of claim 17, wherein said requests are made to said module manager as one of a synchronous or asynchronous request, wherein synchronous requests are handled on a first-in-first-out basis, and wherein asynchronous requests are processed and returned when completed.

21. (new) The computer readable medium of claim 17, further comprising computer executable instructions for performing the steps of:

creating an instance of said module manager upon receiving said request; and

discarding said instance after said response has been handled.

22. (new) A module manager that manages a request for information received at a mailbox, comprising:

a registry of information modules;

a module loading function for dynamically loading said information modules upon receipt of said request,

wherein said module manager routes said request to an appropriate information module for resolution, and wherein said appropriate information module resolves said request and returns a response to said module manager.

23. (new) The module manager of claim 22, wherein said module manager maintains a list of supported services provided by each of said information modules.

24. (new) The module manager of claim 23, wherein said module managers handles service collisions such that if plural information modules register as supporting a same service, said module manager determines which of said plural information modules will handle said request.

25. (new) The module manager of claim 22, wherein said module loading function includes local and remote module loading functions, wherein said local loading function loads information modules that reside on a same physical device as said module manager, and wherein said remote loading function load functions that reside on devices logically connected to said module manager.

26. (new) The module manager of claim 25, wherein local modules communicate with said module manager via one of memory calls, object inheritance, and inter-process communication.

27. (new) The module manager of claim 25, wherein remote modules communicate with said module manager via TCP/IP sockets.

28. (new) The module manager of claim 22, wherein said request is made as one of a serializable Java object, XML placed in an HTTP header, or an XML-RPC-enabled web server.

29. (new) The module manager of claim 28, wherein said request is either synchronous or asynchronous, wherein a synchronous request is handled on a first-in-first-out basis, and wherein an asynchronous request is processed and a response returned in accordance with a processing time of the request.

30. (new) The module manager of claim 22, wherein instances of said module manager are created each time a new request is received and discarded after the request has been handled.

31. (new) The module manager of claim 22, further comprising a user interface, wherein said user interface is adapted to configure said module manager.